Updated Final Review Draft, 9/25

TERADATA®

What would you do if you knew?™

Teradata Virtual Machine Community Edition

Installation, Configuration, and Upgrade Guide

Release 15.01

B035-5946-095K

September 2015

TERADATA. LABS

The information contained in this document is provided on an "as-is" basis, without warranty of any kind, either express or implied, including the implied warranties of merchantability, **fitness** for a particular purpose, or non-infringement. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you. In no event will Teradata Corporation be liable for any indirect, direct, special, incidental, or consequential damages, including lost **profits** or lost savings, even if expressly advised of the possibility of such damages.

The information contained in this document may contain references or cross-references to features, functions, products, or services that are not announced or available in your country. Such references do not imply that Teradata Corporation intends to announce such features, functions, products, or services in your country. Please consult your local Teradata Corporation representative for those features, functions, products, or services available in your country.

Information contained in this document may contain technical inaccuracies or typographical errors. Information may be changed or updated without notice. Teradata Corporation may also make improvements or changes in the products or services described in this information at any time without notice.

To maintain the quality of our products and services, we would like your comments on the accuracy, clarity, organization, and value of this document. Please e-mail: teradata-books@lists.teradata.com

Any comments or materials (collectively referred to as "Feedback") sent to Teradata Corporation will be deemed non-confidential. Teradata Corporation will have no obligation of any kind with respect to Feedback and will be free to use, reproduce, disclose, exhibit, display, transform, create derivative works of, and distribute the Feedback and derivative works thereof without limitation on a royalty-free basis. Further, Teradata Corporation will be free to use any ideas, concepts, know-how, or techniques contained in such Feedback for any purpose whatsoever, including developing, manufacturing, or marketing products or services incorporating Feedback.

Table of Contents

Preface

Purpose

This guide contains steps to install, configure, and use Teradata Virtual Machine Community Edition (Community Edition).

To install Community Edition, you must be an experienced VMware administrator.

Audience

This guide is intended for use by:

- VMware Administrators who configure vSphere and configure and deploy Community Edition templates.
- Database and software professionals who use the deployed Community Edition instances, ensure the database is running, and use client applications to access the database for evaluation.

Revision History

Date	Description
September 2015	Initial Release.

Additional Information

Related Links

URL	Description
www.teradata.com	External site for product, service, resource, support, and other customer information.
www.info.teradata.com	External site for published Teradata customer documentation.
https://tays.teradata.com	External site for access to the Teradata software server. Accessible only with an active service contract.

Product Safety Information

This document may contain information addressing product safety practices related to data or property damage, identified by the word *Notice*. A notice indicates a situation which, if not avoided, could result in damage to property, such as equipment or data, but not related to personal injury.

Example

Notice: Improper use of the Reconfiguration utility can result in data loss.

CHAPTER 1
Community Edition Overview

Community Edition Description

Community Edition is a SUSE Linux Enterprise Server (SLES) operating system and Teradata Database packaged into a virtual container that runs in a virtualized environment on third-party hardware.

Community Edition software consists of a template and associated property files and scripts.

When a Community Edition virtual machine is deployed, it operates as a fully functional instance of the configured Teradata Database.

Note: You must have administrative privileges on the destination VMware environment to install and configure Community Edition virtual machines.

Community Edition Use Cases

Once deployed by the VMware administrator, Community Edition can be used to evaluate Teradata Virtual Machine and Teradata Database.

CHAPTER 2

Preparing for Teradata Virtual Machine Installation

Requirements

Community Edition has the following technical requirements.

Component	Requirements
VMware ESXi version	• ESXi 5.5 or 6.0
Datastore space for template deployment	• Teradata Database: 30 GB minimum
vCenter management	• VMware administrator privileges • Running vCenter Management instance (verified using the Microsoft Services control within the vCenter image) • vCenter Standard version (must support ESXi 5.5 or 6.0 servers) • vCenter 5.5 or 6.0
vSphere Client	• Windows OS for vSphere Client • vSphere PowerCLI version 5.5 release 2 or version 6.0
PowerCLI	• Windows .Net Framework 4.5 • Windows PowerShell V2 or latest • vSphere PowerCLI toolkit

Teradata Virtual Machine Installation Components

Teradata Virtual Machine installation components consist of a template, property files, and scripts.

- The template contains the Teradata database and underlying operating system.
- Property files define configuration settings for the overall Teradata Virtual Machine environment and the specific virtual machines.
- Scripts deploy the Teradata virtual machine based on the configuration information in the property files.

Teradata Virtual Machine Template

Template	Template Name	Default **Configuration**
Teradata Database and Operating System Template	SLES 11: • TVME-TD15.00__SLES11_ESXi.15.00.xx.xx-1 • TVME-TD14.10__SLES11_ESXi.14.10.xx.xx-1 SLES 10: • TVME-TD15.00__SLES10_ESXi.15.00.xx.xx-1 • TVME-TD14.10__SLES10_ESXi.14.10.xx.xx-1	• 2 virtual cores • 8 GB RAM • Four 20 GB virtual data disks • 1 network adapter

Teradata Virtual Machine Property Files

Properties File Name	Contents
`vmwareit.properties`	Virtual environment configuration settings common across Teradata virtual machines. Define the values in this file once, unless something in the virtual environment changes.
`deploy.properties`	Configuration settings unique to the Teradata virtual machine. Define the values in this file for each deployment.

Teradata Virtual Machine Scripts

Script	Description
`deploy.ps1`	Parses the `vmwareit.properties` and `deploy.properties` files to ensure that all required values are present and correctly formatted, validates settings, and launches each subsequent script to deploy the components of the Teradata virtual machine.
`TVMCEDeployAndConfigureDBS.ps1`	Deploys the Teradata Database and underlying operating system for the Teradata virtual machine.

Installing Teradata Virtual Machine

You must be a VMWare administrator to install and configure Teradata Virtual Machine software.

1 Log on to Teradata Developer Exchange Downloads.

2 Download and save the Community Edition templates and scripts.

3 If applicable, limit administration of the Teradata Virtual Machines host server to certain users.

4 Set PowerCLI Script execution privileges.

5 Edit the `vmwareit.properties` file.

6 Edit the `deploy.properties` file with the values for the specific Teradata Virtual Machine installation.

7 Run the setup script to install the Teradata virtual machine.

8 Confirm that the Teradata Database installed correctly.

9 If you plan to use Teradata QueryGrid, configure the connector.

Specifying Administrators of a Teradata Virtual Machine Host Server

To allow only certain users to administer a Teradata Virutal Machine host server in a shared virtual environment, add those users as administrators of a datacenter containing the server.

1 Open vSphere Client.

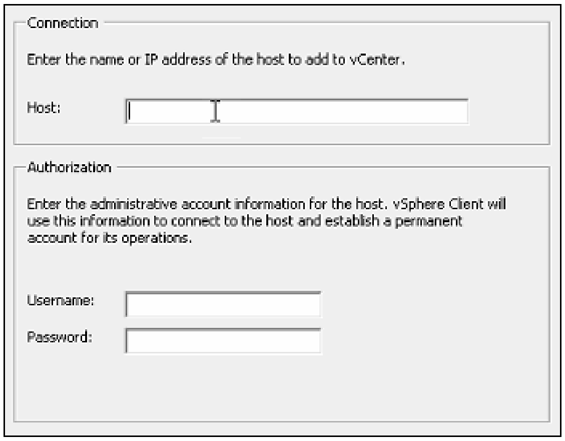
2 Log in to the vCenter Server associated with the Teradata Virtual Machine host server.

3 In the tree view, right-click the vCenter Server listing, and select New Datacenter.

4 In the tree view, click New Datacenter, and rename it as appropriate.

5 Add the ESXi server that will host the Teradata virtual machines to the new datacenter:

a In the tree view, right-click the datacenter branch, and select Add Host.



Enter the name or IP address of the host to add to vCenter.

Enter the administrative account information for the host. vSphere Client will use this information to connect to the host and establish a permanent account for its operations.

b At Host, enter the name or IP address of the ESXi host.

c At Authorization, enter the username and password, and click Next.

d If an authenticity message appears, click Yes to continue.

e At Host Summary, click Next.

f At Assign License, ensure that the Essentials license is selected, and click Next.

g Click Next through the remaining prompts, then click Finish.

The host name appears in the tree view, under the datacenter branch.

6 In the tree view, click the branch for the datacenter containing the Teradata Virtual Machine host server.

7 Click the Permissions tab.

8 In a blank area of the tab, right-click, then select Add Permission.

The Assign Permissions window appears.

9 Under Assigned Role, select Administrator.

10 Under Users and Groups, click Add.

The Select Users and Groups window appears.

11 From the Domain list, select the Active Directory where the accounts for the users who you want to designate as administrators exist.

12 In the list pane, select the applicable users or groups.

13 Click Add.

14 When you have finished selecting users or groups, click OK.

The Select Users and Groups window closes, and the Permissions tab reflects the selected users or groups.

Setting PowerCLI Script Execution Privileges

If this is a new installation of PowerCLI, you must set the PowerCLI script execution privileges to allow the unsigned but Teradata-certified Teradata Virtual Machine scripts to be run.

1 In a PowerCLI command window, run: Set-ExecutionPolicy -ExecutionPolicy unrestricted

2 If you are prompted to confirm the change, enter `Y`.

Obtaining the Latest Installation Components

Before installing Teradata Virtual Machine, download the latest templates, property files, and scripts from Teradata Developer Exchange.

1 Log on to Developer's Exchange: Teradata Developer Exchange Downloads

2 Under Display Downloads From, select Database.

3 In the download list, select Community Edition, and download the `CommunityEdition.rar` and `Template.rar` files.

4 Decompress the downloaded `.rar` files.

Editing Property Files

Edit the `vmwareit.properties` file only once unless something in your virtual environment changes. Edit the `deploy.properties` file before deploying each Teradata virtual machine.

1 In a text editor, open the properties file you need to edit.

The files are in `/CommunityEdition/`.

2 Update the file as needed.

3 Save the file.

Related Topics

The vmwareit.properties File, on page 13

The deploy.properties File, on page 14

The vmwareit.properties File

The `vmwareit.properties` file contains the VMware vSphere environment and IT configuration settings common to every Teradata virtual machine deployment.

Configure the values in the `vmwareit.properties` file only once unless something in the virtual environment changes.

Property Category	Properties	Description
VMWare Details	`vCenterServerNameOrIPAddress`	The name or IP Address of the vCenter server. Example: `vCenterServerNameOrIPAddress=`*`192.168.0.0`*

vCenter Server Credentials	`vCenterServerUserName`
	`vCenterServerPassword`
		Credentials required if you want to use the `-force` option for unattended deployment; otherwise, you must manually provided the credentials when prompted during deployment.
		Example:
		`vCenterServerUserName=`*`Admin`*
		`vCenterServerPassword=`*`Password`*

Common Network Details	`NetworkDomain`
	`NetworkPrimaryDns`
	`NetworkSecondaryDns`
		The network domain and primary DNS (both required) and the secondary DNS (optional).
		Example:
		`NetworkDomain=`*`corp.company.com`*

Property Category	Properties	Description

		`NetworkPrimaryDns=`*`192.168.0.1`*
		`NetworkSecondaryDns=`*`192.168.0.2`*

Network Switch vSwitch0	`VirtualNetworkPortGroupName`	The physical adapter with port group for the virtual network. The default value is `VM Network`. Example: `VirtualNetworkPortGroupName=`*`VM Network`*

The deploy.properties File

The `deploy.properties` file contains the specifications needed by the deployment scripts to install and configure a given Teradata Database virtual machine.

Set the values in the `deploy.properties` file for each Teradata Virtual Machine deployment.

Property Category	Properties	Description
VMWare Host Details	`ESXiHostName`	Name of the VMware host on which the virtual machine will run. Example: `ESXiHostName=`*`esxhost01.corp.company.com`*
Datastore	`Datastore`	Datastore associated with the host. Example: `Datastore=`*`datastore001-vol01`*

VMWare Template Details	`DeployDBSOVFTemplate`
	`DatabaseOVFFilePath`
	`ExistingDBSTemplateName`
		Source of the template for the virtual machine being deployed in the specified VMWare environment.
		To use an OVF template from your local computer, set `DeployDBSOVFTemplate` to `true` and specify the `DatabaseOVFFilePath` value.
		Example:
		`DeployDBSOVFTemplate=true`
		`DatabaseOVFFilePath=C:\\TVME\\TVME-TD15.00__SLES10_ESXi.15.00.02.02-1.ovf`
		To use a template that was previously loaded to the vCenter environment, set `DeployDBSOVFTemplate` to `false` and specify the `ExistingDBSTemplateName` value.
		Example:
		`DeployDBSOVFTemplate=false`
		`ExistingDBSTemplateName=TVME-TD15.00__SLES11_ESXi.15.00.03.05-1`

Property Category	Properties	Description

VM Name and Location	`DbsVMName`
	`LocationName`
		The name of the virtual machine and the folder to which it should be deployed in the VMware environment.
		Examples:
		`DbsVMName=`*`TVME-CE1`*
		`Location=`*`FolderName`*

DBS Configuration	`NumberOfDataDisks`
	`DataDiskSizeGB`
		Total number of data disks and size of each data disk. The number of data disks must be even, and the data disk size must be an increment of 10 GB. The maximum database size (`NumberOfDataDisks` X `DataDiskSizeGB`) is 1024 GB.
		Examples:
		`NumberOfDataDisks=4`
		`DataDiskSizeGB=20`

VM Configuration	`NumCPU`
	`MemoryGB`
		Number of CPUs and amount of memory. Allocate at least 2 GB of memory for each data disk.
		Examples:
		`NumCPU=2`
		`MemoryGB=8`

DBSControl Flags	`Temporal`
	`Columnar`
	`BlockLevelCompression`
	`RowLevelSecurity`
	`SQLH`
	`QueryGrid`
	`QueryGridVersion`
		Database features, enabled by setting the value to `true`. Accept the default of `false` if you do not plan to use the feature.
		Note: If you plan to enable Temporal and restore data from a different Teradata Database, review Knowledge Article KAP1BACEE, *Migrating to TimeDateWZControl* first. This scenario can result in subtle data quality issues which are outlined in the Knowledge Article.
		If enabling QueryGrid, also specify the `QueryGridVersion` value. For Teradata Database version 14.10, use `SQLH` instead of `QueryGrid` for this feature, and do not specify the `QueryGridVersion` value.
		Note: QueryGrid is supported for HDP 1.3 and 2.1 versions.

Static IP Address and Host Name for Database VM	`IPAddressSetting`
	`HostNameForDbsVM`
	`IPAddressForDbsVM`
	`SubnetmaskForDbsVM`
	`NetworkGatewayForDbsVM`
		The IP address setting and the host name, IP address, subnet mask, and network gateway for the Teradata Database virtual machine.
		• The default host name is `TVME-CE`.
		• If `IPAddressSetting=dhcp` (the default), only the `HostNameForDbsVM` property value is required; if `IPAddressSetting=static`, all other property values are required.
		• The `subnetmask` value must be in the format of `XXX.XXX.XXX.X`.
		Examples:

Property Category	Properties	Description

		`IPAddressSetting=static`
		`HostNameForDbsVM=`*`TVME-CE`*
		`IPAddressForDbsVM=`*`192.168.0.10`*
		`SubnetmaskForDbsVM=`*`255.255.254.0`*
		`NetworkGatewayForDbsVM=`*`192.168.1.254`*

Unattended Deployment

Teradata Virtual Machine allows unattended deployment using a command line parameter. The unattended deployment uses login credentials provided in the `vmwareit.properties` file, and assumes the default response to each prompt normally shown during deployment.

To deploy Teradata Virtual Machine without manual intervention:

- Specify the vCenter login credentials in the `vmwareit.properties` file.
- Configure the other values in the `vmwareit.properties` and `deploy.properties` files.
- Run the deployment script with a -force parameter.

CHAPTER 3

Teradata Virtual Machine Deployment

Executing the Setup Scripts

The start-up script, `deploy.ps1`, verifies that all required property file values are present and correctly formatted, and validates other settings. If it finds no errors, the script deploys the virtual machine.

If you use the `-force` parameter, the script runs unattended. See Unattended Deployment for more information.

1 Open the PowerCLI console.

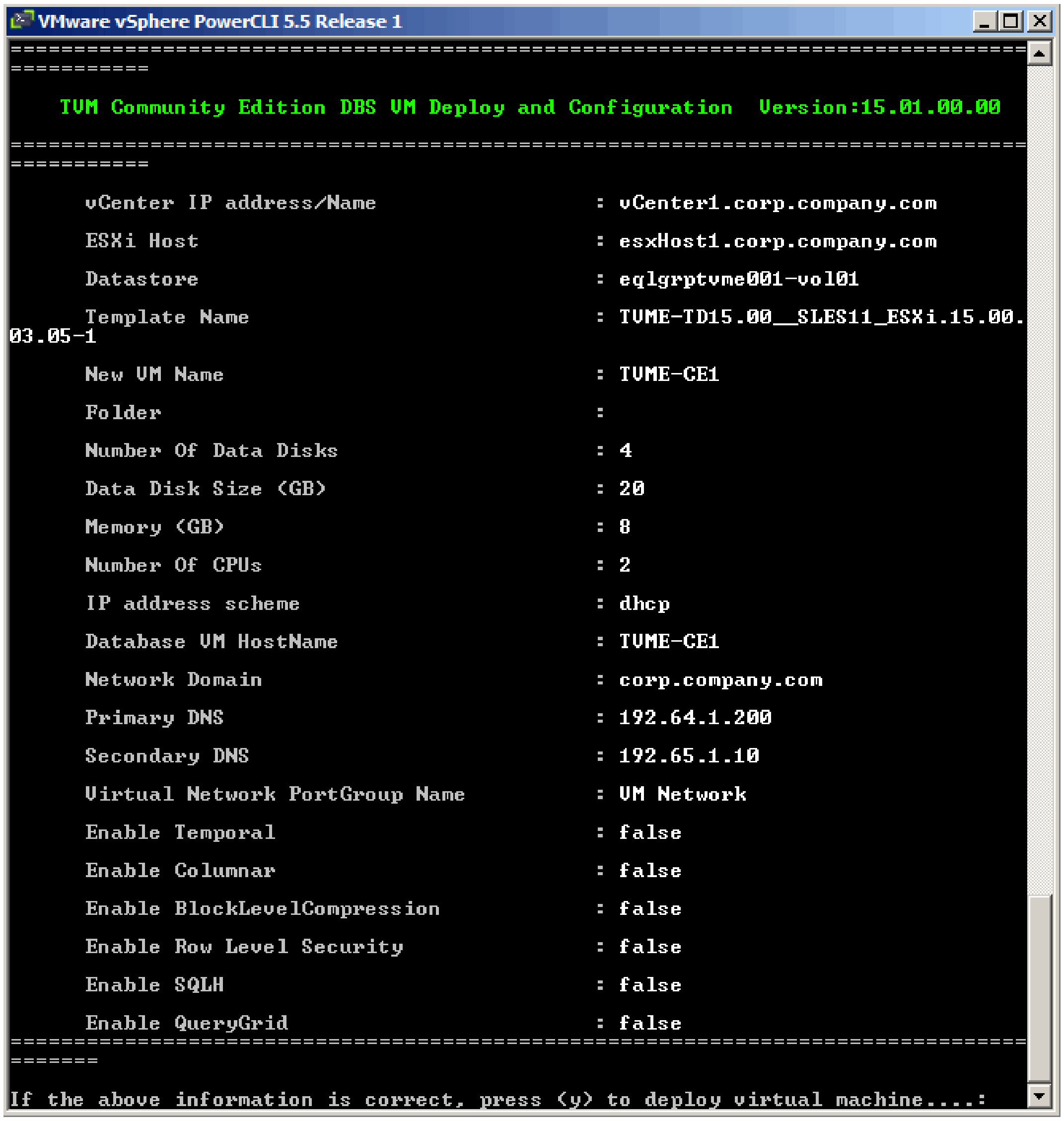
2 Change the directory to the location where the downloaded Teradata Virtual Machine installation files are stored.

3 Run the deploy.ps1 script:

```
.\deploy.ps1 -vmitFile vmwareit.properties -tvmeFile
deploy.properties
```

Note: For unattended deployment, use the `-force` parameter: `.\deploy.ps1 -force -vmitFile vmwareit.properties -tvmeFile deploy.properties`

The script validates all the entries in the `vmwareit.properties` and `deploy.properties` files.



```
================================================================================
===========

    TVM Community Edition DBS VM Deploy and Configuration  Version:15.01.00.00

================================================================================
===========

      vCenter IP address/Name                      : vCenter1.corp.company.com

      ESXi Host                                    : esxHost1.corp.company.com

      Datastore                                    : eqlgrptvme001-vol01

      Template Name                                : TVME-TD15.00__SLES11_ESXi.15.00.
03.05-1

      New VM Name                                  : TVME-CE1

      Folder                                       :

      Number Of Data Disks                         : 4

      Data Disk Size (GB)                          : 20

      Memory (GB)                                  : 8

      Number Of CPUs                               : 2

      IP address scheme                            : dhcp

      Database VM HostName                         : TVME-CE1

      Network Domain                               : corp.company.com

      Primary DNS                                  : 192.64.1.200

      Secondary DNS                                : 192.65.1.10

      Virtual Network PortGroup Name               : VM Network

      Enable Temporal                              : false

      Enable Columnar                              : false

      Enable BlockLevelCompression                 : false

      Enable Row Level Security                    : false

      Enable SQLH                                  : false

      Enable QueryGrid                             : false
================================================================================
=======

If the above information is correct, press <y> to deploy virtual machine....:
```

4 Review the information. If it is correct, press y.

If the information is not correct, edit the property files and re-run the script.

The system prompts you to log on to the vCenter server.

5 Type your vCenter User name and Password, then click OK.

The script connects to the vCenter server and validates the information in the property files.

When validation is successful, the script deploys the selected template, and configures the virtual machine based on the values in the property files.

6 If a message warns you that running scripts is disabled, you must set the script execution privileges, then re-run the script.

See Setting PowerCLI Script Execution Privileges.

Deployment Verification

After a Teradata Virtual Machine deployment, test the Teradata Database.

Testing the Database

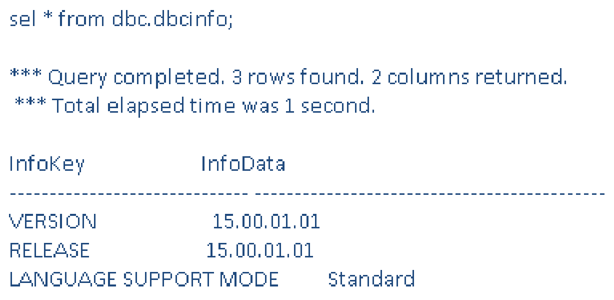
After a Teradata Virtual Machine deployment, verify that the database is accessible and responsive.

1 Start BTEQ.

2 Using the default credentials, log in to the database.

3 Run select * from dbc.dbcinfo to see if the database is responding.

Sample result indicating response:



```
sel * from dbc.dbcinfo;

*** Query completed. 3 rows found. 2 columns returned.
 *** Total elapsed time was 1 second.

InfoKey                  InfoData
----------------------------- -------------------------------------------
VERSION                   15.00.01.01
RELEASE                  15.00.01.01
LANGUAGE SUPPORT MODE        Standard
```

4 If the database does not respond, try to access the server through SSH.

5 If you cannot access the server, contact Teradata Customer Service.

APPENDIX A

Installing Teradata QueryGrid

Teradata QueryGrid Installation Overview

If you configured a Teradata virtual machine to enable the Teradata QueryGrid feature, you must take additional steps to complete installation of the feature after deploying the virtual machine. The steps differ depending on the version of Teradata QueryGrid you are using. Supported versions are as follows:

- Teradata Database-to-Teradata Database (TD-to-TD)
- Teradata Database-to-Hortonworks Data Platform (TD-to-HDP)

Teradata Database-to-Teradata Database Installation

To complete installation of the TD-to-TD version of Teradata QueryGrid on a virtual machine, perform the outlined procedures in order.

Acquiring Remote Proxy User Information

When installing Teradata QueryGrid Teradata Database-to-Teradata Database, a Teradata proxy user must be configured. The proxy user is the user used to log on between Teradata systems and impersonates the local user on the remote Teradata Database system.

For configurations with Teradata Database-to-Teradata Database on both local and remote Teradata systems, it is recommended that a unique proxy user be created for systems in both sides of the configuration.

1 Confirm the proxy user values have been provided.

Adding IP Addresses to the Teradata Host File

Resolve all hostname conflicts and confirm the hostnames are not being resolved through a local DNS.

1 On all Teradata TPA nodes, save a copy of the `/etc/hosts` file at:

```
cp /etc/hosts/etc/orig.hosts
```

2 Add the Teradata TPA nodes IP addresses to `/etc/hosts`.
For example:

- `192.168.135.100 node002-8`
- `192.168.135.101 node002-9`

Configuring Teradata Proxy User Setup

For configurations where TD-to-TD is desired on both the local and remote Teradata systems, a unique proxy user name for each system is recommended.

The proxy user usually has no perm space or permissions beyond acting as proxy. However, the user may require SPOOL and TEMPORARY space based the specific query workload of all users of the remote server. Adjust as appropriate.

1 Confirm the remote proxy user exists.

2 On the remote Teradata Database system, log in as Administrator.

For example, log in as user *dbc*.

3 Run the following query:

```
CREATE USER proxyuser AS PERM = 0 PASSWORD = password;
```

Granting User Privileges

The Teradata Database user requires specific privileges to run the installation setup. These privileges can be granted by the database administrator. If the administrative user is not known, use *dbc*.

Note: GRANT statements are not required if using an administrator such as *dbc* to run the setup.

1 Grant the execute procedure on SQLJ:

```
GRANT EXECUTE PROCEDURE ON SQLJ TO user WITH GRANT OPTION;
```

2 Grant the create procedure on SYSLIB to user:

```
GRANT CREATE PROCEDURE ON SYSLIB TO user WITH GRANT OPTION;
```

3 Grant the create external procedure on SYSLIB:

```
GRANT CREATE EXTERNAL PROCEDURE ON SYSLIB TO user WITH GRANT OPTION;
```

4 Grant the drop procedure on SYSLIB:

```
GRANT DROP PROCEDURE ON SYSLIB TO user WITH GRANT OPTION;
```

5 Grant the drop function on SYSLIB:

```
GRANT DROP FUNCTION ON SYSLIB TO user WITH GRANT OPTION;
```

6 Grant the execute function on SYSLIB:

```
GRANT EXECUTE FUNCTION ON SYSLIB TO user WITH GRANT OPTION;
```

7 Grant the create function on SYSLIB:

```
GRANT CREATE FUNCTION ON SYSLIB TO user WITH GRANT OPTION;
```

8 Grant select on SYSLIB:

```
GRANT SELECT ON SYSLIB TO user WITH GRANT OPTION;
```

9 Grant all on user to user:

```
GRANT ALL ON user TO user, dbc WITH GRANT OPTION;
```

10 Grant UDTTYPE on SYSUDTLIB:

```
GRANT UDTTYPE ON SYSUDTLIB TO user WITH GRANT OPTION;
```

11 Grant the create external procedure on TD_SERVER_DB:

```
GRANT CREATE EXTERNAL PROCEDURE ON TD_SERVER_DB TO user WITH GRANT
OPTION;
```

12 Grant the drop procedure on TD_SERVER_DB:

```
GRANT DROP PROCEDURE ON TD_SERVER_DB TO user WITH GRANT OPTION;
```

13 Grant the execute function on TD_SERVER_DB:

```
GRANT EXECUTE FUNCTION ON TD_SERVER_DB TO user WITH GRANT OPTION;
```

14 Grant the create function on TD_SERVER_DB:

```
GRANT CREATE FUNCTION ON TD_SERVER_DB TO user WITH GRANT OPTION;
```

15 Grant select on TD_SERVER_DB:

```
GRANT SELECT ON TD_SERVER_DB TO user WITH GRANT OPTION;
```

16 Grant create server on TD_SERVER_DB:

```
GRANT CREATE SERVER ON TD_SERVER_DB TO user WITH GRANT OPTION;
```

17 Grant drop server on TD_SERVER_DB:

```
GRANT DROP SERVER ON TD_SERVER_DB TO user WITH GRANT OPTION;
```

18 Grant insert on TD_SERVER_DB:

```
GRANT INSERT ON TD_SERVER_DB TO user WITH GRANT OPTION;
```

19 Grant select on TD_SERVER_DB to SYSLIB:

```
GRANT SELECT ON TD_SERVER_DB TO SYSLIB;
```

Creating Foreign Server Object and Accessibility

This procedure defines the remote server and sets accessibility to users on the local system. Perform all steps on the local Teradata Database.

Use this procedure for installations and when creating new remote servers or access. Do not use for upgrades.

1 Grant the proxy user privileges for all users to be impersonated:

```
GRANT CONNECT THROUGH proxyuser TO PERMANENT local_user without
role;
```

2 Create an authorization object for the remote server:

```
CREATE AUTHORIZATION td_server_db.remote_server_hostnameproxy AS
DEFINER TRUSTED USER 'proxyuser' PASSWORD 'password';
```

This creates the proxy in the td_server_db database. The *DEFINER* clause is used to make this global to all users.

3 Log into the local Teradata system and grant DBC access to create server objects in the name of the remote system:

```
GRANT CREATE SERVER ON td_server_db TO dbc;
```

4 Create the server:

```
CREATE FOREIGN SERVER <remote_server_hostname>
EXTERNAL SECURITY DEFINER TRUSTED <remote_server_hostname>proxy
USING
Hosttype('Teradata')
remotehost ('<remote_server_hostname>.<FQDN>')
local_ips('<local_server_hostname>.<FQDN>')
port('5000')
```

Note: Ensure the port is open. The port must be open for validating.

```
read_timeout(200)
listen_timeout(60)
concurrentstreams(1)
DO IMPORT WITH syslib.load_from_td,
DO EXPORT WITH syslib.load_to_td;
```

For example:

```
CREATE FOREIGN SERVER sdll7151
EXTERNAL SECURITY  DEFINER TRUSTED sdll7151proxy
    USING
    Hosttype('Teradata')
    remotehost ('sdll7151.labs.teradata.com')
    local_ips('sdlc0062.labs.teradata.com')
    port('5000')
    read_timeout(200)
    listen_timeout(60)
    concurrentstreams(1)
        DO IMPORT WITH syslib.load_from_td,
        DO EXPORT WITH syslib.load_to_td;
```

5 From database td_server_db;, grant *SELECT* and *INSERT* on the remote server to local users.

```
GRANT SELECT ON remote_server_hostname to local user;
GRANT INSERT ON remote_server_hostname to local user;
```

For example:

```
GRANT SELECT ON sdll7151 to sdlc_user2;
GRANT INSERT ON sdll7151 to sdcl_user2;
```

Validating the Installation

Verify that the proxy user is setup.

1 Log into the local Teradata Database as a local user.

2 To validate the installation run the Help Foreign Server query:
HELP FOREIGN SERVER remote_server_hostname;

Validating the System

Validate the local and remote nodes.

1 On the local and remote nodes configured for Teradata Query Grid:

```
/opt/teradata/gsctools/bin/chk_all
```

2 Review the output and confirm it is correct:

```
/var/opt/teradata/gsctools/chk_all
```

Teradata Database-to-Hortonworks Data Platform Installation

To complete installation of the TD-to-HDP version of Teradata QueryGrid on a virtual machine, perform the outlined procedures in order.

Acquiring Remote Proxy User Information

This topic applies to versions of Teradata-to-Hadoop prior to Teradata 15.00.02. The Hadoop administrator usually performs this procedure.

For Teradata QueryGrid to work with a Hadoop system, a Teradata proxy user must be configured on the Hadoop NameNode. This proxy user must be allowed to access HDFS from the Teradata nodes on behalf of another Hadoop user in a secured way. The Teradata proxy user for this setting is *tdatuser*.

1 Confirm the following:

- The proxy user values have been provided.
- Both *tdatuser* and the *tdatuser* home directory exist in `/home`.

  For example: `/home/tdatuser`
- The default shell for the user is set to `/etc/passwd`.

  For example: `/bin/bash`

2 On the Hadoop side, the following configurations are required in `core-site.xml` to add *tdatuser* as a trusted proxy user:

a Determine the file system groups that *tdatuser* may impersonate.

b Determine the hosts and nodes from where the *tdatuser* user may access the HDFS.

These configurations must be present, otherwise impersonation is not allowed and Teradata queries fail with a security error.

Identifying the HCatalog Server

All nodes in the Hadoop cluster must have the HCatalog libraries installed. HCatalog is the entry point for the Teradata QueryGrid connector.

1. If the customer has changed the default Ambari server login credentials of *admin*/*admin*, obtain the new updated credentials.
2. Log onto Ambari on a Hadoop node:
   - For TDH systems, use port 8081.
   - For HDP systems, use port 8080.

   If an external connection already exists, use SWS through Server Management or the customer name.
3. Click the Hive tab.
4. Click **Configs**.
   The Hive Metastore host is displayed.

Running the System Validation Script

After identifying the HCatalog Server, run the system validation script.

1. Save `sys_val.pl` to the `/home/support` directory on both the Teradata and Hadoop nodes.
2. Verify the system by running `sys_val.pl` on both the Teradata control and PDN nodes and Hadoop master node.

   `sys_val.pl` should be run separately and independently on both types of systems.

Description	Command
Teradata Control or PDN nodes and Hadoop Master nodes	`/home/support/sys_val.pl -cc`
Scripts with CR/LF, binary FTP from Windows	`perl /home/support/sys_val.pl -cc`

Resolving Hostname Conflicts

Teradata-to-Hadoop uses node hostnames to resolve network addressing. However, it is possible that there may be conflicts and duplicate hostnames between Teradata nodes and Hadoop nodes that must be resolved before Teradata QueryGrid is installed.

If problems exist, contact the Teradata Global Support Center, Hadoop Support team for changes.

Setting Up Hadoop Cluster for LDAP

Before using Teradata QueryGrid with LDAP, update the storage format in the LDAP directory.

Note: `tdsqlh_td 15.00.02.01` is the minimum version of the Teradata QueryGrid connector package required to use with LDAP.

1 In Ambari, open the Hive configuration page.

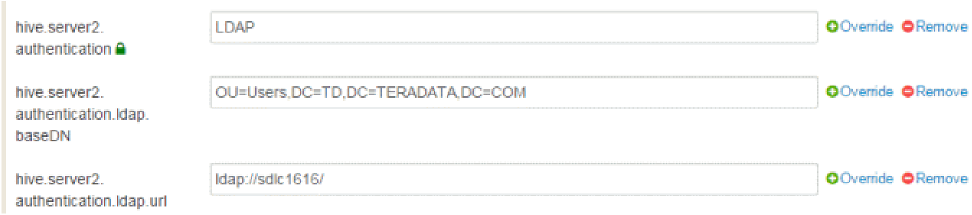
2 Complete the fields:

hive.server2	Description
hive.server2.authentication	LDAP
hive.server2.authentication.ldap.base DN	Matches the directory location where the authenticated users are stored on the LDAP server
hive.server2.authentication.ldap.url	Matches the correct LDAP server

Note: HiveServer2 requires the schema for user Distinguished Names (DN) to follow the format `uid=<username>,baseDN` where:

- *`username`* is the name of the user being added
- *`baseDN`* is the directory where the authenticated usernames are stored

For example:



hive.server2.authentication: LDAP

hive.server2.authentication.ldap.baseDN: OU=Users,DC=TD,DC=TERADATA,DC=COM

hive.server2.authentication.ldap.url: ldap://sdlc1616/

3 Save the changes and restart all Hive services.

Adding Hadoop IP Addresses to the Teradata Host File

Resolve all hostname conflicts and confirm the hostnames are not being resolved through a local DNS.

1 On all Teradata TPA nodes, save a copy of the `/etc/hosts` file at:

```
cp /etc/hosts/etc/orig.hosts
```

2 Add the Hadoop node's IP addresses to `/etc/hosts`.
For example:

- `192.168.135.100 hdp002-8`
- `192.168.135.101 hdp002-9`

3 Confirm the byn1 IP addresses on Hadoop systems using byn1.

Configuring Hadoop for Teradata Proxy Setup

The method used to configure the Teradata proxy user on the Hadoop NameNode is determined by the Hadoop system version.

Configuring Hadoop for Teradata Proxy Setup for Hadoop 2.1 Systems Using Ambari

Note: This topic applies to versions of Teradata-to-Hadoop prior to Teradata 15.00.02.

Use Ambari to edit the `core-site.xml` file. Note the following:

- Property value changes made in Ambari appear in the `core-site.xml` file.
- Property value changes made in `core-site.xml` through manual editing do not appear in Ambari.

If Ambari is used for cluster management, then also use Ambari for modifying service property values.

1. Verify the customer has provided the following information needed for the configuration.
   - Ambari server login and password

     The default Ambari server username and password is *admin*/*admin*. If the customer has changed the Ambari server password, it must be provided prior to the installation.
   - Network access to Hadoop Master Node 1
   - The values to complete the Add Property field have been obtained

2. Log onto Ambari Hadoop Master Node 1:
   - For TDH systems, use port 8081.
   - For HDP systems, use port 8080.

   Use SWS through Server Management or customer name if external connections already exist.

   For example, for HDP systems, use `http://hdp002-1:8080` and for TDH systems, use `http://hdp002-1:8081`
3. Click the Services tab.
4. From the left pane, click HDFS.
5. On the HDFS screen, click the **Configs** tab.
6. Expand Custom core-site.xml.
7. Configure the Teradata proxy user:

   The default wildcard value for these properties is `*` and allows impersonation from any host or user. If specific groups and hosts have been identified, replace * with the groups and hosts in a comma separated list.

   a. Click Add Property and add a property with the key value `hadoop.proxyuser.tdatuser.groups` and value `*`.

   b. Click Add Property and add a property with the key value `hadoop.proxyuser.tdatuser.hosts` and value `*`.

8 Click Save.

9 When the Restart button appears, restart HDFS by clicking Restart > Restart All > **Confirm** Restart All.

Configuring Hadoop for Teradata Proxy Setup for Hadoop 1.3.2 Systems

Note: This topic applies to versions of Teradata-to-Hadoop prior to Teradata 15.00.02.

For Hadoop 1.3.2 systems, Ambari is disabled and you must edit the `core-site.xml` file manually.

1 Navigate to the Hadoop NameNode configuration file:

```
/etc/hadoop/conf/core-site.xml
```

2 Add the following properties to the file: and properties to the file.

- `hadoop.proxyuser.tdatuser.groups`
- `hadoop.proxyuser.tdatuser.hosts`

```
<property>
 <name>hadoop.proxyuser.tdatuser.groups</name>
 <value>users</value>
 <description>
        Allow the proxy user tdatuser to impersonate any members of
HDFS group(s). For example, ‘users’ is used as HDFS group that
tdatuser is allowed to impersonate users belonged to this group.
 </description>
</property>

<property>
  <name>hadoop.proxyuser.tdatuser.hosts</name>
  <value>host1,host2</value>
  <description>
        The proxy user can connect only from host1 and host2 to
impersonate a user. Here host1 and host2 represents Teradata nodes.
All nodes of the Teradata system need to be listed here in order
for SQL-H query to be processed. It is recommended to use the IP
addresses of the Teradata nodes.
   </description>
</property>
```

The property values are based on the Teradata and Hadoop environment setup requirements.

3 Save the `core-site.xml` file.

4 Restart the NameNode:

```
hcli system restart
```

Configuring JVM Heap Size for Teradata Database 15.0

This procedure applies only to Teradata Database 15.00 installations.

`cufconfig` was enhanced for Teradata QueryGrid 15.0. A new option, `JVMOptions`, was added so that the environment file does not have to be placed on all nodes. The options needed can be added using `cufconfig` and are automatically replicated to all nodes.

1 If upgrading, before beginning the upgrade process remove any existing JVM options:

   a On the primary Teradata node, edit `/tmp/jvm_base.txt` by removing `/tmp/jvmopt.txt` from the `JavaEnvFile` field.

   b Run `cufconfig –f /tmp/jvm_base.txt`.

   c Run `cufconfig –o`.

   The `JavaEnvFile` field value must be empty.

2 Calculate the Java Heap values.

   To do the memory calculations, see Knowledge Article: KAP3160E8E, *Teradata QueryGrid Teradata Database-to-Hadoop Orange Book*.

3 In the `/tmp` directory of the primary Teradata node, create a new `jvmopt.txt` file with the following options:

```
JVMOptions: -server -XX:UseParallelGC -XX:+UseParallelOldGC –
Xms7100m -Xmx7100m -XX:NewSize=2370m -XX:MaxNewSize=2370m -
XX:MaxPermSize=864m
```

4 Set the `jvmopt.txt` file permissions:

```
psh chmod 777 /tmp/jvmopt.txt
```

5 Run `cufconfig –f /tmp/jvmopt.txt`.

6 Run `cufconfig –o`

   The `JVMOptions` field value must be populated.

7 Restart the Teradata Database:

```
tpareset -y restart-with-sqlh
```

Validating the Installation

Validate the Teradata and Hadoop setups are ready for Teradata-to-Hadoop queries.

1 Create the `hcatalog` table with data:

   a On the primary Teradata node, navigate to `/opt/teradata/sqlh/`*version*.

   b Download `tdsqlh_example.hive` and `tdsql_data.csv` and copy them to the `/tmp` directory on the Hadoop NameNode.

   c Log into the Hadoop NameNode and navigate to the `/tmp` directory.

   d Change files permissions on the copied files:

```
chmod 777 tdsqlh_example.hive tdsql_data.csv
```

   e Change the `user` to *Hive*:

```
su hive
```

2 Run a sample Hive script by creating a `tdsqlh_test` table with 14 columns and 805 populated rows:

```
hive < tdsqlh_example.hive
```

Verify the script completes and returns row count as 805.

```
Total MapReduce CPU Time Spent: 4 seconds 580 msec
OK
805
Time taken: 33.76 seconds
```

3 Use SQL Assistant or BTEQ to log into the Teradata primary node as user *dbc*.

4 Run the Teradata-to-Hadoop query to import rows from the `tdsqlh_test` table. Use the following values:

5 Run the query to count the rows in the `tdsqlh_test` table. If the server hostname is unknown, look in the `/etc/hosts` file.

- For TDH systems, *('hdp001-hive')* is used in the following example to specify the server. TDH systems use the Hive VIP when specifying the server.
- For HDP systems, *('hdp001-1')* is used in the following example to specify the server. HDP systems use the hostname of the master node when specifying the server.

For Hadoop 2.1 systems with one master node and Hadoop 1.3.2 systems, run the following SQL query:

```
SELECT count(*)
FROM SYSLIB.load_from_hcatalog(USING
    server('hdp001-hive') or ('hdp001-1')
    port('9083')
    username('hive')
    dbname('default')
    tablename('tdsqlh_test')
    columns('*')
    templeton_port('50111')

hadoop_properties(‘dfs.client.use.datanode.hostna
me=t
rue’)
) as D1;
```

For Hadoop 2.1 systems with two or more master nodes, make the required replacements, then run the following SQL query:

- Replace the `dfs.nameservices` name ("YOURCLUSTER" in the example) with the defined cluster name.
- Replace the `dfs.namenode.rpc-addresses` ("NAMENODE1" and

```
SELECT count(*)
FROM SYSLIB.load_from_hcatalog(USING
    server('hdp001-hive') or ('hdp001-1')
    port('9083')
    username('hive')
    dbname('default')
    tablename('tdsqlh_test')
    columns('*')
    templeton_port('50111')

hadoop_properties(‘<dfs.client.use.datanode.hostn
ame=<true>,
<dfs.datanode.use.datanode.hostname=true>,
<dfs.nameservices=<YOURCLUSTER>,
<dfs.ha.namenodes.YOURCLUSTER=nn1,nn2>,
<dfs.namenode.rpc-
address.YOURCLUSTER.nn1=NAMENODE1:8020>,
```

"NAMENODE2" in the example) with the defined addresses.	<dfs.namenode.rpc- address.YOURCLUSTER.nn2=NAMENODE2:8020>, <dfs.client.failover.proxy.provider.YOURCLUSTER=o rg.apache.hadoop.hdfs.server.namenode.ha.Configur edFailoverProxyProvider>')) AS D1;

Clustername and namenode rpc-addresses are located in the `hdfs-site.xml` settings: Services > HDFS > **Configs** > Custom hdfs-site.xml

The following are common terms found in this script:

Term	**Definition**
Server	DNS hostname or IP address for the Hadoop NameNode
Port	Port for the Hadoop NameNode service
templeton_port	The web `Hcatalog` port

If the query returns an error instead of row count 805 then the Teradata-to-Hadoop setup requires manual troubleshooting to isolate the problem.

Validating the System

1 To verify the system, independently run the following on both the Teradata Control/PDN node and Hadoop master node:

`sys_val.pl`

Description	Command
Teradata Control or PDN nodes and Hadoop Master nodes	`/home/support/sys_val.pl -cc`
Scripts with CR/LF, binary FTP from Windows	`perl /home/support/sys_val.pl -cc`